ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

UDC 621.3.01+621.38

DOI: https://doi.org/10.30837/ITSSI.2022.20.022

P. Klimushyn, T. Solianyk, O. Mozhaiev, Y. Gnusov, O. Manzhai, V. Svitlychny

CRYPTO-RESISTANT METHODS AND RANDOM NUMBER GENERATORS IN INTERNET OF THINGS (IOT) DEVICES

Subject of research: crypto-resistant methods and tools of generating random sequences and hardware support of cryptographic transformations in IoT devices. The **aim** of the article is to study crypto-resistant methods and tools for generating and testing random sequences suitable for use in IoT devices with limited resources; determination of circuit implementations of random sequences hardware generators; formation of conclusions on the use of random number generators (RNG) in cryptographic protection systems of the IoT network. The article solves the following **tasks:** analysis of methods and hardware for generating random sequences to protect IoT solutions with limited resources; identification of safe and effective technologies for the implementation of RNG; classification of RNG attacks; analysis of the shortcomings of the practical use of statistical test packages to assess the quality of random sequences of RNG; evaluation of the speed of cryptoaccelerators of hardware support for cryptographic transformations; providing practical guidance on RNG for use in resource-constrained IoT devices. Research **methods:** method of structural and functional analysis of RNG and IoT devices, cryptographic methods of information protection, methods of random sequence generation, method of stability analysis of systems, methods of construction of autonomous Boolean networks and Boolean chaos analysis, methods of quality assessment of random sequences. **Results** of work: the analysis of technologies and circuit decisions of hardware RNG on characteristics: quality of numbers' randomness and unpredictability of sequences, speed, power consumption, miniaturization, possibility of integral execution; providing practical recommendations for the use of RNG in cryptographic protection systems of the IoT network. The **novelty** of the study is the analysis of methods and hardware to support technologies for generating random sequences in the system of cryptographic protection of IoT solutions; classification of attacks on RNG and features of protection against them; identification of effective RNG technologies and circuit solutions for use in low-power IoT devices with limited computing resources; providing practical recommendations for the use of RNG in cryptographic protection systems of the IoT network. The analysis of technologies and circuit solutions allowed to draw the following **conclusions:** protection of IoT solutions includes: security of IoT network nodes and their connection to the cloud using secure protocols, ensuring confidentiality, authenticity and integrity of IoT data by cryptographic methods, attack analysis and network cryptographic stability; the initial basis for the protection of IoT solutions is the true randomness of the formed RNG sequences and used in algorithms for cryptographic transformation of information to protect it; feature of IoT devices is their heterogeneity and geographical distribution, limited computing resources and power supply, small size; The most effective (reduce power consumption and increase the generation rate) for use in IoT devices are RNG exclusively on a digital basis, which implements a three-stage process: the initial digital circuit, normalizer and random number flow generator; Autonomous Boolean networks (ABN) allow to create RNG with unique characteristics: the received numbers are really random, high speed – the number can be received in one measure, the minimum power consumption, miniature, high (up to 3 GHz) throughput of Boolean chaos; a promising area of ABN development is the use of optical logic valves for the construction of optical ABN with a bandwidth of up to 14 GHz; the classification of known classes of RNG attacks includes: direct cryptanalytic attacks, attacks based on input data, attacks based on the disclosure of the internal state of RNG, correlation attacks and special attacks; statistical test packages to evaluate RNG sequences have some limitations or shortcomings and do not replace cryptanalysis; Comparison of cryptoaccelerators with cryptographic transformation software shows their significant advantages: for AES block encryption algorithm, speeds increase by 10-20 times in 8/16-bit cryptoaccelerators and 150 times in 32-bit, growth hashing of SHA-256 in 32-bit cryptoaccelerators more than 100 times, and for the NMAS algorithm - up to 500 times.

Keywords: Internet of Things; random number generator; cryptocurrency; cryptanalysis; cryptographic keys; encryption; hashing; Autonomous Boolean Networks; Boolean chaos; statistical tests; cryptoaccelerators.

Introduction

The Internet of Things (IoT) is a major IT development trend that contributes to societal development in the areas of human life services, public safety, medical services, industrial process management, transportation, competitiveness and business productivity. The interconnection of IoT nodes connected to the network negatively affects the overall level of security and crypto-resistance of the system. This problem is exacerbated by factors of mass, heterogeneity in the structure of the devices involved, limited computing and energy resources, the automation of connecting IoT nodes to the network, gullibility from the users and protection of their personal data and privacy. Problems of security are a major restraining factor in the use of IoT technologies.

Protecting IoT solutions includes securing IoT network nodes and their connection to the cloud using secure protocols, confidentiality, authenticity and integrity of data during transmission, processing and storage on the IoT network, as well as resistance to physical and virtual attacks. Main properties of information (confidentiality, authenticity and integrity) are preserved by cryptographic methods, such as data encryption and data hashing using cryptographic keys.

High-performance information systems are manufactured on general-purpose microprocessors with significant processing power, large memory capacity and power-intensive supply. Along with this, for embedded IoT systems the computing power, power consumption, size and price are limited. There is a problem of implementation of widespread cryptographic algorithms (AES, SHA, HMAC, RSA, DH, ECC, etc.) in embedded IoT devices.

In turn, the generation of cryptographic keys is based on the unpredictability of random sequences of numbers generated by LFO. That is, the initial basis for the protection of IoT solutions is the true randomness of LFO number sequences. Therefore, the problem of information protection in IoT devices is related to the simultaneous optimization of the level of security of HFSCs, their productivity and price under the above-mentioned constraints. In addition, a characteristic of IoT devices is their heterogeneity and geographical distribution. Consequently, the generation of random sequences by software using a deterministic algorithm in high-performance microprocessor systems does not ensure the proper quality of these sequences. Such sequences are pseudorandom, which means that the possibility of their predictability grows, and, as a result, the crypto resistance of the systems decreases.

To overcome the above-mentioned problems, a wide range of technical solutions for generating truly random sequences using hardware HFOs using physical sources of unexpected noise, including quantum processes or Boolean chaos in digital circuits is implemented. Specialized hardware components (cryptoaccelerators) are created in general-purpose microcontrollers of different families, significantly accelerating random sequence generation and execution of existing cryptoalgorithms. Such specialized modules are effective for securing IoT solutions and are a critical trend in IoT deployments in resource-constrained environments.

Analysis of the literature. Analysis of scientific and technical literature shows that in recent decades a large number of elementary pseudorandom number generators, which include linear congruent generators, Fibonacci generators with delay [1], was developed and investigated. Many years of research led to the conclusion that all of them are not crypto-resistant and can be part of the shapers of pseudorandom sequences. The most successful such generators in terms of cryptography are considered in detail in the work of Bruce Schneier [2]. Over time, algorithms that seemed previously reliable find new weaknesses. For this reason, in the process of development of cryptographic protocols there is a question of finding new engineering solutions to build effective crypto-resistant generators, free of the identified weaknesses.

In contrast to other engineering problems, the development of a new random number generation algorithm is different in that a reliable answer to the question about the effectiveness of the found solution may appear some time later, when an individual cryptanalysis method is developed for it. The developer can be satisfied only with results of preliminary testing of the solution by existing test packages [3-5].

The crypto resistance of generators is a key factor for the whole system of information protection by means of cryptographic transformations. The impact of this factor on the cryptocurrency of the protection system is studied in a large number of works, in particular [6, 7], etc.

The best choice for forming truly random sequences is hardware implementation of generators. These include generators using physical processes in electronic elements [8, 9], quantum generators [10-12], digital generators based on autonomous Boolean networks [13-16]. A general trend in the development of hardware generators is the creation of specialized modules in integral design [9]. Implementation of such modules is effective for low-power IoT devices.

Thus, the analysis of scientific sources showed a huge range of means of generating random sequences. IoT network security requires crypto-resistant, low-power, miniaturized RNGs.

The **purpose** of this article is to study crypto-resistant methods and means of generating and testing random sequences suitable for IoT devices with limited resources; definition of schematic implementations of the hardware random sequence generators; formation of conclusions for the use of RNG in cryptographic network protection systems of IoT devices.

1. Crypto-resistant methods of random number generation

Random number sequences are used for:

- generation of session, public and secret cryptographic keys in symmetric and asymmetric encryption systems, electronic signature systems;
- random data sets in IoT identification and authentication protocols.

The quality of cryptographic key formation and random datasets determines the cryptographic stability of the entire IoT network as a whole.

Methods of generating keys and cryptographic data are divided into two classes: random (based on physical processes using hardware) and pseudorandom (based on software, mathematical software). The tools that generate pseudorandom number sequences are called pseudorandom number generators (PRNG) [17].

In hardware RNG, each bit of raw data is based on an unanticipated physical process, that is, it is produced from the noise signal of the internal source of physical analog noise. The value of a random number obtained directly by discretizing the analog noise signal.

PRNGs are implemented programmatically according to the algorithm, starting from an initial value that can be generated by a hardware entropy source. Due to the deterministic nature of the processing, it is assumed that PRNGs produce pseudorandom, not random bits. The initial number used to implement PRNG must contain sufficient entropy to guarantee randomness [18]. For them, the generation of new random numbers does not increase the entropy of the initial number.

Nowadays RNG based on different physical processes, such as electric current noise in electric elements (resistors, diodes, transistors), radioactive decay, atmospheric turbulence, cosmic radiation, photoelectric effect, quantum phenomena have been developed.

Disadvantage of the most typical random number generators based on physical processes is the emergence in the process of generation of so-called shifted sequences (in such sequences a certain combination of numbers or bits is most often repeated). Offset occurs because of the difficulty in designing and implementing precisely balanced physical schemes of number generation. To remove such a disadvantage there are algorithms for further processing.

2. Effective technologies for implementing hardware random number generators

Modern RNG in IoT devices are based on physical principles and implemented on electronic components. They use, for example, such physical processes as thermal noise in electronic elements or avalanche breakdown of p-n junction with reverse bias in Zenner diode [8]. The disadvantage of such generators is the need to manufacture a special analog circuit, which greatly complicates the manufacture of integrated circuits and makes it impossible to implement in a programmable logic integrated circuit (PLIC).

In addition, RNG are known which use the difference in frequency of two generators, caused by thermal drift. For example, the chip Intel 82802 has two generators (fast and slow), which are located in different parts of the chip and measure the difference in their frequency. Such RNG can be implemented entirely by means of digital logic, because as a rule, inverters with feedback and a chain of buffer elements as a delay line are used as generators. The disadvantages of such generators are:

- the need to place generators at a distance from each other in the chip topology (to reduce the thermal correlation between them);
- low performance (since it is necessary to accumulate the drift for some time of operation);
- not high quality (unpredictability) of random numbers.

In addition, this generator is difficult to implement in FPGA, because automatic development tools do not allow to control the physical placement of generators from each other at a safe distance to reduce the frequency correlation between them.

One of the most crypto-resistant methods of obtaining random numbers in nature are quantum RNGs. In quantum solutions, the entropy of a bit is considered a priori equal to one, taking the position of absolute unpredictability of the value of that bit. Decreasing of quantum bit entropy results from an admixture of classical deterministic chaos. Absolute randomness of a finite sequence may be compromised by a faulty source.

The principle of quantum RNG is based on generating single photons by a light source and directing them to a translucent mirror. The photon may reflect, or it may pass through the translucent mirror with equal probability fates. The choice a photon makes is completely random. At the output of the system are counters that record the number of photons passed and repelled [12].

Most light sources emit photons at random points in time and the number of photons released per unit time is quite random. This fact is the basis of quantum RNGs, built on the basis of a light-sensitive matrix based on CMOS technology of an ordinary camera by a group of scientists from the University of Geneva under the leadership of Bruno Sanguinetti. Each pixel of the matrix counts the number of photons falling on its surface during a certain period of time. These photons are converted into electrons by the corresponding multiplier of the light-sensitive matrix. The number of electrons during the same period will differ by a completely random number [10].

The problem with generators using physical processes is that their analog circuit wastes energy. In addition, it is difficult to keep this analog circuit working because of improvements in the technical process to produce the chips and to miniaturize them. Therefore, it is important to have a fully digital circuit that allows the microprocessor to generate a rich stream of random values without these problems.

The first such RNG solution was proposed in [12], consisting of a circuit with undefined states, which can be in a certain state of logical zero or one. Of course, a digital circuit can detect short periods of time in an undefined state by switching between these two logical values. However, it must work clearly and must never generate between them; otherwise, it would cause delays or even failure in the system [11].

This digitized approach to random discharge generation would work fine if all inverter circuits were absolutely identical. However, the chaos of the physical world never permits this. In fact, no two inverters are exactly alike. The presence of slight differences in the speed or strength of their responses can become a nuisance and jeopardize the randomness of the bit sequence. To keep the inverters in equilibrium, a feedback mechanism is built in. This helps to satisfy one of the rules of statistical randomness: a long stream of numbers must have approximately the same number of all possible digits. By tuning the internal operation of each inverter, you can guard against predictability.

Simultaneously with the development of a robust digital random number source, other Intel engineers developed the additional logic needed to efficiently process and deliver these bit sequences. The unclean bitstream coming from the underlying hardware, regardless of quality, can have bias and correlation.

To guarantee the quality of random numbers, a three-step process is developed that involves an initial digital circuit, a normalizer, and an initial sequence shaper [11]. This three-step random number generation process prevents any predictability variations.

This generator generates a stream of random numbers at a rate of 3 Gb/s. That is, it allows to get rid of the inconvenience of analog components, which are used in physical generators to measure random physical processes, significantly reduce power consumption and increase the rate of generation by more than 30 thousand times. [11].

For the last few years the so-called "autonomous Boolean networks" (ABN) have been actively studied. Such a network is a topologically connected graph of logical elements, to which no control or tact signals are submitted from outside [13; 16].

At the same time, such a network is subject to obvious additional requirements:

- no element can have "hanging" (not connected to any output) inputs;

- no two outputs must be connected to each other.

These requirements are due to the certainty of the state of the logical elements of the network and the safety of its operation. Depending on the topology, the ABN may exhibit different behavior:

- to be in a stable or quasi-stable state;
- generate with a certain frequency and waveform;
- to be in a state of so-called "Boolean chaos".

An example of the simplest ABN is a repeater with the output connected to the input. Such a network is in a quasi-stable state: depending on the initial state (0 or 1) it will stay in it indefinitely. Another example is one or more inverters connected to each other in a ring. Depending on the number of inverters in the ring, their behavior will be stable, quasi-stable or oscillating.

One inverter is likely to be in a stable state between 0 and 1. This is due to the physical implementation of the inverter as a high gain amplifier with negative feedback equal to 1. For the same reason the lack of generation in very small networks with elements with low output slew rate is possible. With an odd number of inverters, the network will generate with a period proportional to the number of inverters. With an even number of inverters the network is in a quasi-stable state. Obviously, the random number generator implemented in the chip Intel 82802 on two generators and measuring the frequency drift between them, in fact, is a special case of ABN - it uses two rings with a different odd number of inverters. More complex networks can often produce chaotic behavior.

There is a direct physical analogy in content that helps to clearly understand the behavior of the ABN in various situations. It is a pendulum. A simple pendulum, when deflected and released, begins to generate with a stable period - generator behavior. If we flip the pendulum upward, the upper point is a quasi-stable state: at any small deflection, the pendulum will fall to the left or right. If we divide the arm of a pendulum into two parts connected by a hinge, we get a so-called chaotic pendulum whose evolution over time cannot be accurately predicted because any infinitesimal change in the initial state (up to the quantum level) increases with time and leads to quite large changes in the behavior of the pendulum. By this analogy, it is possible to measure random deviations in the frequency of two pendulums arising due to chaotic external influences, but this requires a long analysis, while it is possible to deflect a chaotic pendulum and obtain a random value in the shortest time.

Thus, oscillating ABNs can indeed be used to generate random numbers, but ABNs that initially exhibit chaotic behavior are much more promising. The rate of onset of Boolean chaos and its qualitative characteristics are determined by a number of factors. The most important characteristic is the Lyapunov exponent. If its exponent is negative, the deviations fade with time. If it is greater than zero, the random deviations are exacerbated by the system. If it is zero, the deviations do not fade or intensify, but accumulate if they enter the system from the outside. For rapid physical generation of random numbers, the index must be positive. A simple pendulum has a negative Lyapunov exponent. This is expressed in the fact that the frequency of oscillation is momentarily stabilized due to the absence of external influences.

A chaotic pendulum is a separate example of a system with a positive exponent, so that any small influence leads to a completely different dynamic in a short time. In binary logic, there are no logic functions that amplify small deviations, but there are functions that allow the changes to disappear. For the case with two arguments these are the XOR (exclusive OR) and XNOR (equivalence) functions. Any change in any input signal in them causes a change in the output signal, so these functions are mostly used in RNG.

In binary logic, there are no logical functions that amplify small deviations, but there are functions that do not allow changes to disappear. For the case with two arguments these are the functions XOR (excluding "or") and XNOR (equivalence). Any change in any input signal in them causes a change in the output signal, so these functions are mostly used in RNG.

In modern science, ABN circuits are known to exhibit chaotic behavior. Chaos occurs when the operation of the network is determined by the smallest deviations in the supply voltage, the landslide fronts through thermal fluctuations, surges and other factors that destabilize the network. In the literature [13] two and three input XOR or XNOR logic elements are considered, in which the output is brought back to the inputs via two (or three) delay lines.

In such networks, chaotic behavior can be observed with a certain ratio of delay line lengths. The problem of the network is the strong dependence of the behavior not only on the ratio of delay line lengths, but also on their physical implementation. In this case, instead of chaos, oscillations may occur. In addition, despite the apparent schematic simplicity of the device, it requires a large number of logic elements, because each delay line is a chain of inverters. As a result, a seemingly simple circuit can contain several dozen elements. Moreover, three out of four variants of such an generator have a fundamental disadvantage - the disappearance of the generation after some time. The XOR element comes to a stable state with zero at the output regardless of the number of inputs, and the XNOR element with two inputs comes to a stable state with one at the output. An XOR element with three inputs has an additional stable state with one at the output.

In [16] it is shown that the construction of a true random number generator, containing a digital chaotically oscillating autonomous Boolean network as a source of entropy, should be focused on providing an increase in the rate of number generation while reducing the energy consumption by constructing such partially controlled ABN, which at minimum size is guaranteed and with maximum possible speed enters the Boolean chaos state. The most important requirement should be the impossibility of stable, quasi-stable or oscillating state of the network.

The rate of chaos growth depends directly on the size of the cyclic signal propagation paths available in the network. The larger the path size, the longer it takes for the signal to propagate through it in order to return to the starting point. This means that the network must have a minimum size. This is obvious from the point of view of an generator consisting of an odd number of inverters: the period of oscillation is directly proportional to the length of the ring.

To address these issues, it is better to consider a synchronous network. In fact, the difference between the synchronous Boolean network and ABN is that the signal at the outputs of all logical elements changes simultaneously, so we can assume that the network goes through a series of states.

It was shown in [13] that mutual influence of different network fragments can lead to their forced synchronization. This is a very important effect, which must be excluded or minimized. Since it is necessary to find the minimum possible network, network variants can be analyzed in order of increasing number of logical elements. A network of at least three elements is needed for chaotic behavior. In general, these must be two or three input elements, since the presence of an element with one input turns the network into a network with two elements. To find the necessary network, you can immediately discard all networks with the presence of a stable state. Also immediately excluded are all networks in which the cycle has less than eight states, since in such networks oscillations and cross-effects are possible.

This leaves only the networks evolving through all eight states. A total of 7! = 5040 different cycles is possible, starting at state 000, going through all possible states and through eight iterations back to state 000. These networks simultaneously have a record low level of complexity. For only eight of them, it is possible to modify the nature of the generation so that they satisfy all of the above requirements and have circuit options. They all consist of three logic elements: two XOR elements (block 4) or XNOR elements (block 5) and one initial three-input element with a more complex, special function called "unit count - block 3" (figures 1., 2., 3. accordingly)[16].

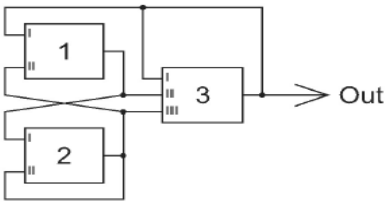

Fig. 1. Functional scheme of the autonomous Boolean network

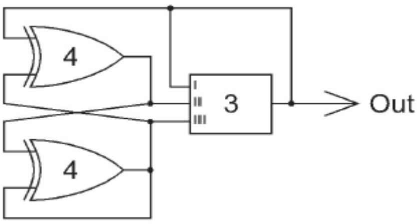

Fig. 2. Logic scheme of an autonomous Boolean network using XOR elements

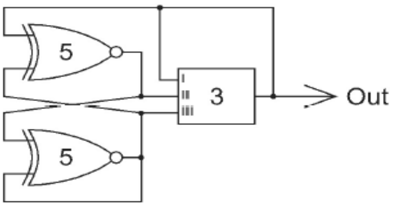

Fig. 3. Logic scheme of an autonomous Boolean network using XNOR elements

Each of the described Boolean networks is a basic unit, on which the random number generator is built. Let's call this basic block "chaotic generator". It has output and modulation input. To reduce the mutual correlation of the signal, the chaotic signal must be removed from the unit count element.

ABN is always in a state of chaotic oscillation and consumes energy in the process. To stop the generation, it is necessary to change the network so that at any initial state it is guaranteed to come to a single possible deterministic state. For the specified networks this cannot be achieved by turning off only one logical element. You need to turn off at least two elements and in the best case these are the same input elements XOR or XNOR. Network can be turned off by forcing the outputs of these elements either to the value 0 or to the value 1. In this case, if the outputs of the XOR/XNOR elements are set to 0, the network output will be set to the value 1, and vice versa.

The chaotic generator itself cannot be used as a random number generator, because it has only an asynchronous output, on which there is a broadband chaotic analog signal. function. On the one hand, it provides a stable initial logic signal, on the other hand, it preserves the previous state, which, if desired, can be used via modulation inputs to produce the next random number.

It should be noted that in spite of mutual modulation, the units of a synchronous chaotic generator can have a shift in the distribution between the number of zeros and ones at the output, i.e. the appearance of one type of quantity at the output is more likely than the other. This is also due to the peculiarities of the physical operation of the logical parts of the circuit. To eliminate this landslide, a procedure called "whitening" of the obtained random numbers may be required.

Thus, ABNs allow you to create RNGs with unique characteristics:

1. The resulting numbers are truly random, which allows them to be used for cryptographic purposes.

2. The rate of random number generation is so high that the network behavior is unpredictable already for the time of signal propagation through several logical elements. That is, when implemented in microprocessor systems, a random number can be obtained in one clock cycle. Such a generation speed actually satisfies any possible needs.

3. the modulation input allows to further improve the characteristics, because another random number can be fed to it. This forces the network to start from a new state each time.

4. The same input allows cross modulation of the discharges of the proposed generator, thereby increasing the rate of chaos buildup.

5. The minimum size of the network makes the proposed generator the most economical in terms of power consumption.

6. The considered generator can be implemented equally effectively on discrete elements, as well as in an integrated design.

7. The generator design is simple and the cost of its implementation is low, which allows it to be used to

support cryptography in devices of low resource and energy-saving devices of the Internet of Things.

It should be noted that ABNs require external synchronization to update the device state, and their state spaces are finite and discrete. Meanwhile, ABNs have continuous state spaces, fast time scales, and complex dynamics. ABNs are systems whose future behavior is determined by past times and transient states. Thus, they can be used in various applications such as random number generation [19], genetic schemes [20], etc. In addition, current systems generating Boolean chaos are usually based on discrete logic elements and have a Boolean chaos bandwidth up to 1.4 GHz [15], and in integrated designs up to 3 GHz [11]. Therefore, due to not the fastest electronic signal processing speed, the bandwidth of Boolean chaos remains limited, which has a direct impact on the prospects of ABN applications.

A promising direction of ABN development is the use of optical logic gates to build optical ABNs. Various groups of manufacturers have already proposed a solution to build optical logic gates based on semiconductor optical amplifiers. Such circuits offer the key advantages of high nonlinear coefficients and easy integration. However, so far they have been used only for information processing in optical communication systems, and not for broadband chaos and complex signals.

In [21] we propose an optical ABN device based on optical logic gates, consisting of an optical logic gate XNOR with two feedback links. By adjusting the delay difference between the two channels the system outputs an optical logic chaotic signal. Such logical chaotic signals with high bandwidths solve the problem of Boolean chaos, which can significantly expand the application area of ABM. The paper [21] describes in detail the physical mechanism of optical Boolean chaos and discusses the effects of feedback delay time, carrier optical amplifier recovery time, and power detection. Embedded integration of the optical Boolean chaotic system can be realized through a combination of optical logic gate integration technology and optical delay line integration technology. Simulation results show that this structure can generate not only periodic, controlled rectangular signals, but also chaotic optical Boolean signals with bandwidths up to 14 GHz.

3. Classification of attacks on random number generators

To strengthen security systems and effectively protect cryptosystems, an important scientific and technical task is to analyze various methods of RNG attack and determine countermeasures to eliminate the consequences of such attacks.

An RNG attack aims to reveal generator parameters for further random number prediction. A successful RNG attack can compromise the cryptographic security of the entire IoT network. Therefore, the use of poor quality RNGs simplifies the complexity of attacks. RNGs in cryptographic systems must meet the following requirements: effective hardware and software implementation; true random statistical sequence, cryptographic attack resistance.

Based on the analysis performed in fig. 4. the most common classification of known classes and types of attacks on RNG [22].

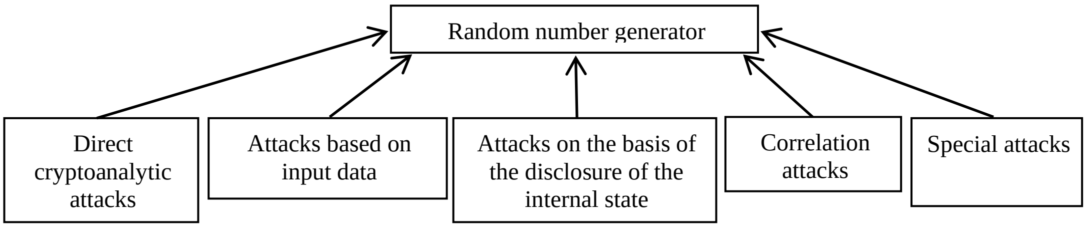
Random number generator
Direct cryptoanalytic attacks
Attacks based on input data
Attacks on the basis of the disclosure of the internal state
Correlation attacks
Special attacks

Fig. 4. Classification of attacks on random number generators

The direct cryptanalytic attacks are connected with tracing the initial RNG data and investigating the patterns of its occurrence. These include partial precalculation and temporal attacks. The partial precalculation attack is used based on the analysis of the internal states of the generator and the observation of consistent raw data. The temporal attack is implemented through tracking the pattern of different amount of time in the change of generator states [23].

Input-based attacks are divided into reproducible input attacks, known input attacks and selected input attacks. Known input attacks are implemented by observing input devices or by input entropy number. Attacks with selected inputs are based on the manipulation of generator inputs. Attacks with reproducible input data are based on the same data as the types of this class are considered [24].

Attacks based on internal state disclosure are possible in four variants: permanent compromise attack (based on the disclosure of RNG state at a point in time), return attack (performed in order to restore the RNG states), iterative guessing attack (uses knowledge of RNG state at a point in time and its intermediate) outputs in order to know the state at the next point in time) and "meet in the middle" attack (is essentially a combination of iterative guessing attack and return attack). This class of attack is successful when the initial state of the RNG is known or provided [23].

Correlation attacks are based on the detection of correlation relationships in the original generator sequence. These attacks are the most common attacks [24]. They include the following attacks: fast correlation attack; basic correlation attacks; attacks based on the use of convolutional and turbo codes; attacks based on the recovery of linear polynomials.

The fast correlation attack based on the parity-check decoding method is the most complex. The basic correlation attack requires high computational complexity.

Convolutional and turbo code attacks ask for huge memory overheads. An attack based on linear polynomial reconstruction is considered the most common.

The class of *special attacks* includes: algebraic attacks, analytical attacks, statistical attacks and repetitive attacks. Algebraic attacks analyze vulnerabilities in the internal algebraic structure. Analytical attacks are based on the detection of structural weaknesses in generators. Statistical attacks are based on comparing the number of 0 and 1 values in a sequence. When performing a repetition attack, the attacker intercepts certain data and then sends it out again, blowing it out as legitimate information. The defense against repetition attacks is the use of sequence numbers and time stamps.

Thus, the analysis of attacks on RNG shows that the attacks have different complexity of their implementation, and therefore require different length of time, computing and other capabilities. At the same time, the details of attack protection should be considered depending on their types, schemes of attack implementation and algorithms of RNG functioning. Here are possible variants of protection against attacks by hashing the initial values of the generator and the initial values of entropy source, periodically changing the internal state of RNG.

4. Testing and evaluating the characteristics of random number generators

RNGs play an important role in the generation of cryptographic keys, initial values in IoT network node authentication schemes, etc. The quality of RNG performance determines the cryptographic strength of the data transmitted in the network [6]. Hence the problem of assessing the quality of random bit sequences RNG, the choice and formation of a system of tools for such verification. Today there is a variety of such tools. In recent decades, the most widely used sets of statistical test packages: D. Knuth, Crypt-X, Diehard, U01; NIST[5].

The first set of statistical tests was the one proposed by D. Knuth. It is based on calculating the value of a statistical criterion and comparing it with tabulated statistical results. The conclusion about the quality of a random sequence is probabilistic. The advantage of tests is their small number and the existence of fast execution algorithms, and the disadvantage is the probabilistic nature in interpreting the results.

Crypt-X statistical tests were developed by researchers at the Australian Security Research Center under a commercial license. The tests support key generators, stream ciphers, and block ciphers. They provide tests for frequency characteristics, sequences of equal bits, linear complexity, sequence complexity, binary derivative, and variable point.

Diehard tests were developed by George Magsaglia to measure the quality of a random number set and are considered one of the most rigorous test sets. This set includes 13 tests. The basis of the tests is to compare the characteristics of the random number sequence of the generator according to the provided specification with the expected values. The disadvantage of the tests is the lack of methods for interpreting their results, heuristic and formation of the evaluation result on a two-point "yes" or "no" scale.

Large library of statistical tests set U01 is implemented in C language. It includes classical tests and some original tests.

The specification and library of the NIST test suite were developed by the US Institute of Standards and Technology in the C language and is considered as a standard [4], which includes 15 tests for the analysis of RNG bit sequences. The feature of the tests is the openness of algorithms and unambiguous interpretation of test results. NIST standard includes requirements and methods of a technological nature and is aimed at solving the problem of statistical quality control of pseudorandom sequences.

Thus, certain statistical test packages have a ready-made software implementation. Their use to evaluate random RNG sequences gives an appropriate level of confidence in their quality. Along with this, the following trends can be noted: the existing number of tests does not provide the solution to the RNG estimation problems from all sides; there are no tests that can be recommended to solve most problems; it is impossible to get a clear conclusion about the randomness of a sequence; all test packages have some limitations or drawbacks. Consequently, the problems of assessing the randomness of RNG sequences are far from being complete and require additional research and improvement of existing approaches. The following issues remain problematic: large length of sequences is required for evaluation; it is impossible to change the parameters of existing test implementations; two-point evaluation ("yes"/"no") of the test result; different language of software packages for testing.

It should be noted that any of the suggested tests, or even the whole package of tests, does not replace cryptanalysis. In this case, preliminary testing is mandatory. A generator, which does not satisfy the testing conditions, is unsuitable. Each of the tests included in the package is focused on searching for a certain type of anomalies in the stream of generated symbols.

The use of application test packages faces a number of serious obstacles. The first one is that they are designed to evaluate already ready generators. In practical work, such devices are developed in stages, gradually bringing them to the level of compliance. The second problem is that each of the tests, which are included in the proposed package, is based on a rather complex theoretical justification, requiring from developers a serious mathematical training and knowledge of various incompatible sections of mathematics. Unfortunately, developers, as a rule, do not give such justification in the instructions attached to the tests. Finally, the third problem is that, although the software provides free access, it is difficult to use it. Most tests involve pre-creation of a file into which a tested pseudorandom sequence is written in the form of 32-bit words, and then the test procedure is run. This is not always convenient and not suitable for all tests, as it requires significant hardware and software resources. In addition, the proposed tests are designed for a particular hardware and software platform.

The listed problems force developers if not to develop their own tests, then to create their own software to implement them. It is convenient to work and can be effectively used in the process of searching for the developed design solution of the generator. Of particular interest are studies on short sequences to obtain adequate results, which is typical for RNG of low-power resources of IoT network nodes [5].

To improve the designed circuits of generators, a method for estimating the number of clock cycles to perform certain operations is used. Let us consider the results of such RNG estimation combined with hardware support of basic cryptographic operations (hashing - MD5/SHA-1/HMAC, encryption - AES/DES/TDES) for major microcontroller companies.

Companies manufacturers have begun to introduce hardware RNGs into microcontrollers. The construction of such generators can be divided into four logical levels: 1) the source of entropy of random bits of hardware noise; 2) the level of processing random bits to eliminate statistical defects; 3) block of random number generation with high-speed high capacity; 4) output buffer to read the sequences in accordance with the generator instructions.

For example, STMicroelectronics corporation on the basis of universal core ARM Cortex-M4F (microcontrollers of STM32F4xx family) has integrated RNG module. That is, a microcontroller with hardware support performs basic cryptographic operations (encryption - AES/DES/TDES, geshing - MD5/SHA-1/HMAC), which significantly increases the cryptographic stability of cryptographic transformations. Evaluation of embedded generator shows, that general level of passability of FIPS-140-2 (Federal Information Processing Standard) tests is 85% [3].

This generator (fig. 5.) is implemented on the basis of analog noise source circuit of two independent generators, over the outputs of which is performed the addition operation modulo 2 (XOR). According to this scheme, generation of 32-bit random number is performed by 40 clock cycles of synchronization signal (RNG_CLK). The scheme provides fault control (clocking and entropy) for random value generation with interrupt signal [25]. More detailed test results of cryptographic quality of embedded RNG are given in [26], where positive results on all 15 randomness tests according to FIPS 140-2 requirements in four initial sequence generation modes are given.

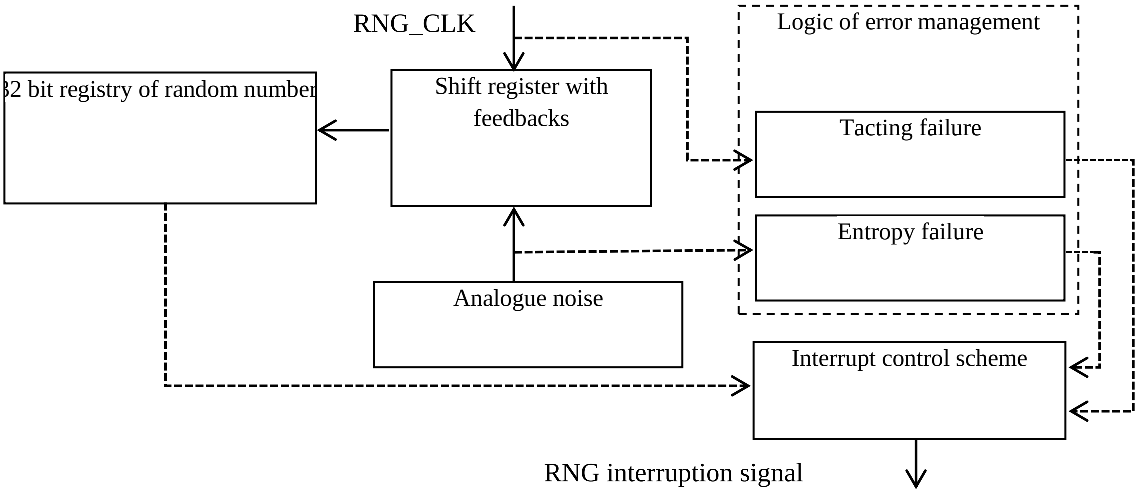

Fig. 5. STM32F2xx/F4xx/F7xx/H7xx RNG microcontroller scheme

In the world of IoT technologies there is a significant security and privacy protection through the development and implementation of general purpose microcontrollers with specialized hardware modules of cryptographic protection of information (cryptoaccelerators). Hardware support for cryptography in IoT devices with cryptoaccelerators allows [26, 28]:

- generate high-quality random number sequences and cryptographic keys;
- store and use cryptographic keys throughout their lifecycle in encrypted form and in secure hardware;
- provide secure personalization (authentication) of IoT devices;
- speed up (tens, hundreds or even thousands of times) the execution of certain crypto-algorithms;
- relieve the CPU from the cryptographic transformation of information;
- provide greater crypto resistance, energy efficiency, and miniaturization of IoT devices.

The core of crypto acceleration engines may include: 1) gas pedals of symmetric encryption based on TDES/AES algorithms, gas pedals of asymmetric cryptography based on RSA and DSA algorithms; 2) gas pedals of hashing based on MD5, SHA-1, SHA-224, SHA-256, HASH algorithms; 3) random number generators based on a physical noise source.

Table 1 shows data from the technical documentation on the number of clock cycles to process one encryption block for the corresponding algorithms: TDES in ECB, CBC modes and AES-128/192/256 in ECB, CBC, CTR, GCM, XTS, GCTR modes. Also in the table to compare performance of cryptoaccelerators for different block sizes with listing by number of clock cycles per byte (cycles per byte - CPB). Besides in the table there is an estimation of performance of cryptoaccelerators in the form of NB parameter as a number of processing bytes for 1 ms at the maximum clock frequency NB = FCPU/(1000×NB).

Comparison of performance of cryptoaccelerators with reproduced means of cryptographic transformations [26] shows their significant advantages. For example, for block AES encryption algorithm performance increases by 10-20 times for 8/16-bit cryptoaccelerators and by 150 times for 32-bit cryptoaccelerators correspondingly. With respect to hash calculation, the performance gain for the SHA-1, SHA-256 algorithms is over 100 times for 32-bit cryptoaccelerators and up to 500 times for NMAS.

In general, there is a trend of hardware support for cryptography in IoT devices with the introduction of comprehensive security solutions, such as identification and authentication of network nodes [27; 28], symmetric and asymmetric encryption, cryptographic protocols, secure key storage and generation, secure application firmware loading and updating, support for digital signatures and certificates, high crypto-resistance and energy efficiency.

Table 1. *Crypto-accelerators speed estimation*

Operation	CPU	Family/model MC	FCPU, MHz	Modes	Tact/ block	CPB, tacts/bytes	NB, byte
Evaluation of crypto-accelerator speed performance of 8-bit microcontrollers							
Enc./Dec. AES-128	AVR	XMega	32	ECB, CBC	375	23,4	1 365
Enc./Dec. AES-128	STM8	STM8L16 STM8AL	16	ECB	892	55,8	287
Enc./Dec. AES-128	i8051	C8051F96x	25	ECB, CBC, CTR	218	13,6	1 835
Enc./Dec. AES-256					298	18,6	1 342
Speed estimation of crypto-accelerators for 16-bit microcontrollers							
Enc./Dec. AES-128	MSP430	MSP430F6xx	25	ECB, CBC, OFB, CFB	167	10,4	2 395
Enc./Dec. AES-128		MSP430FR5x	16		168	10,5	1 524
Enc./Dec. AES-256		MSP430FR6x			234	14,6	1 094
Enc./Dec. TDES	PIC24	PIC24FJ64	32	ECB, CBC, OFB, CFB, CTR	26	3,3	9 846
Enc./Dec. AES-128		PIC24FJ128			219	13,7	2 338
Enc./Dec. AES-256		PIC24FJ256			299	18,7	1 712
Evaluation of crypto-accelerator speeds of 32-bit microcontrollers							
Enc./Dec. TDES	ARM7TDMI	SAM 7XC	55	ECB, CBC, OFB, CFB	50	6,3	8 800
Enc./Dec. AES-128				ECB, CBC, OFB, CFB, CTR	12	0,8	73 333
Enc./Dec. AES-256					14	0,9	62 857
Enc./Dec. AES-128	ARM Cortex-M4F	SAM E5x SAM D5x	120	ECB, CBC, OFB, CFB, CTR, GCM	57	3,6	33 684
Enc./Dec. AES-256					77	4,8	24 935
Hash SHA-1					85	1,3	90 353
Hash SHA-256					72	1,1	106 667
TRNG-32					84	21,0	5 714
Enc./Dec. AES-128	ARM Cortex-M7	SAM E70 SAM S70 SAM V70 SAM V71	300	ECB, CBC, OFB, CFB, CTR, GCM	10	0,6	480 000
Enc./Dec. AES-256					14	0,9	342 857
Hash SHA-1					85	1,3	225 882
Hash SHA-256					72	1,1	266 667
TRNG-32					84	21,0	14 286
Enc./Dec. TDES	ARM Cortex-M7F	STM32H7xx	400	ECB, CBC	64	8,0	50 000
Enc./Dec. AES-128				ECB, CBC,CTR,GCM, CCM, GMAC	14	0,9	457 143
Enc./Dec. AES-256					18	1,1	355 555
Hash SHA-1					82	1,3	312 195
Hash SHA-256					66	1,0	387 879
TRNG-32					54	13,5	29 630

Conclusions

Protection of IoT solutions includes security of IoT network nodes and their connection to the cloud using secure protocols, confidentiality, authenticity and integrity of data during transmission, processing and storage in the IoT network, as well as resistance to physical and virtual attacks. Information is protected by cryptographic methods (encryption and hashing) using cryptographic keys. The initial basis for the crypto resistance of IoT solutions is the true randomness of the sequences formed by the RNG and used in cryptographic transformation algorithms of information for its protection. The peculiarities of IoT devices are their heterogeneity and territorial distribution, limited computing resources and power supply, miniaturization. Due to the above features of IoT, generation of random sequences by software in a deterministic algorithm in high-performance microprocessor systems is unacceptable. Such sequences are of pseudorandom nature, which means that the possibility of their predictability increases, and, as a result, the crypto resistance of the systems decreases.

The problem of generators that use physical processes is that their analog circuit wastes energy. In addition, it is difficult to maintain the performance of this analog circuit due to improvements in the technical process for the production of chips and their miniaturization. Generators on digital circuits can significantly reduce power consumption and increase the rate of generation by more than 30 thousand times. To guarantee the quality of random numbers, they implement a three-step process, which involves an initial digital circuit, a normalizer and a shaper of the initial flow of random numbers.

Autonomous Boolean networks allow to create RNG with unique characteristics: 1) obtained numbers are really random, which allows to use them for cryptographic purposes; 2) speed of random number generation is so high, that when implemented in microprocessor systems random number can be obtained in one clock cycle; 3) modulation input allows to further improve characteristics, that is allows cross modulation of generator bits, thus increasing speed of chaos increase; 4) minimal network size makes the proposed generator the most economical in terms of power consumption; 5) the generator can be implemented equally effectively on discrete elements, as well as in integrated design; 6) the generator design is simple and the cost of its implementation is low, which allows to use it to support cryptography in low-resource and energy-saving IoT devices with a huge bandwidth of Boolean chaos to 3 GHz.

A promising direction of ABM development is the use of optical logic elements to build optical ABMs, the structure of which consists of an optical logic valve XNOR with two feedback links on the optical delay lines. Simulation results show that this structure can generate not only periodic, controlled rectangular signals, but also chaotic optical Boolean signals with a bandwidth up to 14 GHz.

RNG in cryptographic systems must meet the following requirements: effective hardware and software implementation; true random statistical sequence, cryptographic attack resistance. Classification of attacks on RNG includes the following classes: direct cryptanalytic attacks; attacks based on input data; attacks based on internal state disclosure; correlation attacks and special attacks. Protection against attacks is achieved by hashing the initial values of the generator and the initial values of the entropy source, by periodically changing the internal state of the RNG.

The practical use of statistical test packages to evaluate RNG sequences encounters a number of serious obstacles: they are designed to evaluate ready-made generators, although in practice such devices are developed in stages, gradually bringing them to the level of compliance; each of the tests is based on a fairly complex theoretical foundation, requiring from generator developers a serious mathematical training; most tests involve the prior creation of a file in which a tested pseudorandom sequence is written in the form of 32-bit words, and then the test procedure is run, which is not always convenient, because it requires significant hardware and software resources; the proposed tests are designed for a recognized hardware and software platform.

Comparison of performance of cryptoaccelerators with lost cryptographic transformations shows their significant advantages. For example, for the AES block encryption algorithm, performance increases by a factor of 10-20 for 8/16-bit cryptoaccelerators and by a factor of 150 for 32-bit cryptoaccelerators. With respect to hash calculation, the performance gain for the SHA-1, SHA-256 algorithms for 32-bit crypto gas pedals is 100 times greater, and for NMAS up to 500 times greater.

Proposals for further research are focused on identifying technologies and effective solutions for hardware support of cryptography in IoT devices with implementation of comprehensive security solutions, such as identification and authentication of network nodes, symmetric and asymmetric encryption, cryptographic protocols, secure key storage and generation, secure application loading, support of digital signatures and certificates, high crypto-resistance and energy efficiency.

References

1. Donald Knuth (2011), *The Art of Computer Programming*, Volumes 1-4A Boxed Set, Third Edition, Reading, Massachusetts: Addison-Wesley, 3168p.
2. Bruce Schneier (2015), *Applied Cryptography: Protocols, Algorithms and Source Code in C*, 20th Anniversary Edition. March 784 p.
3. FIPS PUB 140-2 (2001), "Security Requirements for Cryptographic Modules", *Federal Information Processing Standards Publication* 140-2, P. 69.

4. Elaine Barker, John Kelsey, (2012), "Recommendation for Random Number Generation Using Deterministic Random Bit Generators", *NIST Special Publication* ,No. 800-90A.
5. Popereshnyak, S.V., Raichev, O.O. (2021), "Research and testing of lightweight pseudo-random number generators for the Internet of Things", *Ukrainian scientific journal of information security*, Vol. 27 (2), P. 71–78.
6. Akram, R. N, Markantonakis, K., Mayes, K. (2012), "Pseudorandom Number Generation in Smart Cards: An Implementation, Performance and Randomness Analysis", *5th International Conference on New Tech- nologies, Mobility and Security (NTMS)*, DOI: https://doi.org/10.1109/NTMS.2012.6208760.
7. Koning Gans G., Hoepman J.-H., Garcia F. D. A. (2008), "Practical Attack on the MIFARE Classic. CARDIS‘08: Proceedings of the 8th IFIP WG 8.8/11.2", *International conference on Smart Card Research and Advanced Applications. Springer*, P. 267–282.
8. Building a Low-Cost White-Noise Generator (2005), "Maxim Integrated. Application note 3469", available at: https://pdfserv.maximintegrated.com/en/an/AN3469.pdf
9. Sovin, Y. R., Nakonechny, Yu. M., Opirsky, I. R., Stakhiv, M. Yu. (2018), "Analysis of cryptography hardware support in IoT devices", *Ukrainian Scientific Journal of Information Security*, Vol. 24, Issue 1, P. 36–48.
10. Anthony Martin, Hugo Zbinden, Nicolas Gisin. (2014), "Quantum random number generation on a mobile phone", available at: http://arxiv.org/pdf/1405.0435v1.pdf
11. Greg Taylor, George Cox. (2011), "Behind Intel’s new random number generator", IEEE Spectrum. Computing, Hardware. September, available at: http://courses.csail.mit.edu/6.857/2012/files/ieee-spectrum.pdf
12. Mario Stipčević, Çetin Kaya Koç (2014), "True Random Number Generators. ResearchGate. Open Problems in Mathematics and Computational Science", P.275–315. DOI: https://doi.org/10.1109/NTMS.2012.620876010.1007/978-3-319-10683-0_12.
13. David P. Rosin, Damien Rontani, Daniel J. Gauthier, and Eckehard Schöll (2013), *Experiments on autonomous Boolean networks*, Chaos 23, 025102.
14. David Rosin. (2015), *Dynamics of Complex Autonomous Boolean Networks*, Doctoral dissertation Universität, Technische Berlin, available at: https://cpb-us-w2.wpmucdn.com/u.osu.edu/dist/7/38882/files/2016/09/rosin_david-14v7bca.pdf
15. R. Zhang, H. L. D. De, S. Cavalcante, Z. Gao, D. J. Gauthier, J. E. S. Socolar, M. M. Adams, and D. P. Lathrop (2009), "Boolean chaos" Phys. Rev. E 80 (4), 045202.
16. Goncharov, S.V. (2021), *Generator of truly random numbers*, Description of the invention to patent RU2741865C1, P. 24. available at: https://patents.google.com/patent/RU2741865C1/ru
17. Gorbenko, Y. I, Shapochka, N.V, Grinenko, T. O, Neyvanov, A. V, Mordvinov, R. I. (2011), "Methods and means of generating pseudo-random sequences", *Applied radio electronics: scientific and technical. Magazine*, Vol. 10. No. 2. P. 141–152.
18. DSTU ISO / IEC 11770-1: 2014, *Information Technology. Methods of protection*, Security key management, Part 1: Structure [to replace DSTU ISO / IEC 11770-1: 2009].
19. D. P. Rosin, D. Rontani, and D. J. Gauthier (2013), "Ultrafast physical generation of random numbers using hybrid Boolean networks", Phys. Rev. E 87 (4), 040902.
20. X. Cheng, M. Sun, and J. E. S. Socolar (2013), "Autonomous Boolean modelling of developmental gene regulatory networks " J. R. Soc., Interface 10 (78), 20120574.
21. Luxiao Sang, Jianguo Zhang, Tong Zhao, Martin Virte, Lishuang Gong, and Yuncai Wang (2020), "Optical Boolean chaos", *Optics Express 29296*, Vol. 28, No. 20/28. available at: https://opg.optica.org/oe/fulltext.cfm?uri=oe-28-20-29296&id=439748
22. Mandrona, M., Garasimchuk, O. (2012), "Attacks on pseudo-random number generators", *Visn. Nat. Lviv Polytechnic University*, No. 741, P. 251–256.
23. Rock, A. (2005), *Pseudorandom Number Generators for Criptographic Applications*, Salzbuburg, 57–65 p.
24. Zenner, E.(2004), *On Cryptographic Properties of LFSR-based Pseudorandom Generators*, Mannheim, 102 p.
25. "Reference manual" (2011), STM32F405xx, STM32F407xx, STM32F415xx and STM32F417xx advanced ARM-based 32-bit MCUs (RM0090), STMicroelectronics, 1316 p.
26. Sovin, Y. R., Nakonechny, Yu. M., Chinka, V. M., Tyshik, I. Y. (2012), "Testing of the built-in random number generator of microcontrollers of the STM32F4XX family according to the NIST STS method", *Lviv Polytechnic National University, Department of Information Protection*, P. 168–175.
27. Klimushin, P., Solianyk, T., Kolisnyk, T., Mozhaev, O. (2021), "Potential application of hardware protected symmetric authentication microcircuitsto ensure the securityof internet of things", *Advanced Information Systems*, Vol. 5, No. 3, P. 103–111.
28. Klimushyn, P., Solianyk, T., Mozhaev, O., Nosov, V., Kolisnyk, T., Yanov V. (2021), "Hardware support procedures for asymmetric authentication of the internet of things", *Innovative Technologies and Scientific Solutions for Industries*, No. 4 (18), P. 31–39.

Received 24.06.2022

Відомості про авторів / Сведения об авторах / About the Authors

Клімушин Петро Сергійович – кандидат технічних наук, доцент, Харківський національний університет внутрішніх справ, доцент кафедри протидії кіберзлочинності, м. Харків, Україна; e-mail: klimushyn@ukr.net; ORCID: https://orcid.org/0000-0002-1020-9399.

Климушин Петр Сергеевич – кандидат технических наук, доцент, Харьковский национальный университет внутренних дел, доцент кафедры противодействия киберпреступности, г. Харьков, Украина.

Petro Klimushyn – Candidate of technical science, associate professor, Kharkiv National University of Internal Affairs, associate professor of Countering Cybercrime Department, Kharkiv, Ukraine.

Соляник Тетяна Миколаївна – кандидат технічних наук, доцент, Харківський національний університет внутрішніх справ, доцент кафедри протидії кіберзлочинності, м. Харків, Україна; e-mail: t.solianyk@khai.edu; ORCID: https://orcid.org/0000-0003-3695-0019.

Соляник Татьяна Николаевна – кандидат технических наук, доцент, Харьковский национальный университет внутренних дел, доцент кафедры противодействия киберпреступности, г. Харьков, Украина.

Tetiana Solianyk – Candidate of technical science, associate professor, Kharkiv National University of Internal Affairs, associate professor of Countering Cybercrime Department, Kharkiv, Ukraine.

Можаєв Олександр Олександрович – доктор технічних наук, професор, Харківський національний університет внутрішніх справ, професор кафедри кібербезпеки та DATA-технологій, м. Харків, Україна; e-mail: mozhaev1957@gmail.com; ORCID: https://orcid.org/0000-0002-1412-2696.

Можаев Александр Александрович – доктор технических наук, профессор, Харьковский национальный университет внутренних дел, профессор кафедры кибербезопасности и DATA-технологий, г. Харьков, Украина.

Oleksandr Mozhaiev – Doctor of technical science, professor, Kharkiv National University of Internal Affairs, professor of Cyber Security and DATA-Technologies Department, Kharkiv, Ukraine.

Гнусов Юрій Валерійович – кандидат технічних наук, доцент, Харківський національний університет внутрішніх справ, завідувач кафедри інформаційних технологій та DATA-технологій, м. Харків, Україна; e-mail: duke6969@i.ua; ORCID: http://orcid.org/0000-0002-9017-9635.

Гнусов Юрий Валерьевич – кандидат технических наук, доцент, Харьковский национальный университет внутренних дел, заведующий кафедрой информационных технологий и DATA-технологий, г. Харьков, Украина.

Yurii Gnusov – Candidate of Technical Sciences, Associate Professor, Kharkiv National University of Internal Affairs, Head of the Department of Information Technologies and DATA-Technologies, Kharkiv, Ukraine.

Манжай Олександр Володимирович – кандидат юридичних наук, доцент, Харківський національний університет внутрішніх справ, завідувач кафедри протидії кіберзлочинності, м. Харків, Україна; e-mail: sofist@ukr.net; ORCID: https://orcid.org/0000-0001-5435-5921.

Манжай Александр Владимирович – кандидат юридических наук, доцент, Харьковский национальный университет внутренних дел, заведующий кафедрой противодействия киберпреступности, г. Харьков, Украина.

Oleksandr Manzhai – Candidate of Legal Sciences,, Associate Professor, Kharkiv National University of Internal Affairs, Head of the Department for Combating Cybercrime, Kharkiv, Ukraine.

Світличний Віталій Анатолійович – кандидат технічних наук, доцент, Харківський національний університет внутрішніх справ, доцент кафедри протидії кіберзлочинності, м. Харків, Україна; e-mail: vit.svet@ukr.net; ORCID: https://orcid.org/0000-0003-3381-3350

Светличный Виталий Анатольевич – кандидат технических наук, доцент, Харьковский национальный университет внутренних дел, доцент кафедры противодействия киберпреступности, г. Харьков, Украина.

Vitaliy Svitlychny – Candidate of Technical Sciences, Associate Professor, Kharkiv National University of Internal Affairs, Associate Professor of the Department of Combating Cybercrime, Kharkiv, Ukraine.

КРИПТОСТІЙКІ МЕТОДИ ТА ГЕНЕРАТОРИ ВИПАДКОВИХ ЧИСЕЛ У ПРИСТРОЯХ ІНТЕРНЕТ РЕЧЕЙ (IOT)

Предмет дослідження: криптостійкі методи та засоби генерування випадкових послідовностей та апаратна підтримка криптографічних перетворень у пристроях IoT. **Метою** статті є дослідження криптостійких методів та засобів генерування та тестування випадкових послідовностей, придатних для використання у пристроях IoT з обмеженими ресурсами; визначення схемних реалізацій апаратних генераторів випадкових послідовностей; формування висновків щодо використання генераторів випадкових чисел (ГВЧ) в системах криптографічного захисту мережі IoT. У статті вирішуються наступні **завдання:** аналіз методів та апаратних засобів формування випадкових послідовностей для захисту рішень IoT з обмеженими ресурсами; визначення безпечних та ефективних технологій реалізації ГВЧ; класифікація атак на ГВЧ; аналіз перешкод практичного використання пакетів статистичних тестів для оцінювання якості випадкових послідовностей ГВЧ; оцінювання швидкодії криптоакселераторів апаратної підтримки криптографічних перетворень; надання практичних рекомендацій щодо RNG для застосування в пристроях IoT з обмеженими ресурсами. **Методи** дослідження: метод структурно-функціонального аналізу RNG та пристроїв IoT, криптографічні методи захисту інформації, методи генерування випадкових послідовностей, метод аналізу стійкості систем, методи побудови автономних булькових мереж та аналізу булевого хаосу, методи оцінювання якості випадкових послідовностей. **Результатами** роботи є аналіз технологій та схемних рішень апаратних ГВЧ за характеристиками: якість випадковості чисел та непередбаченість послідовностей, швидкодія, енергоспоживання, мініатюрність, можливість інтегрального виконання; надання практичних рекомендацій щодо для застосування ГВЧ в системах криптографічного захисту мережі IoT. **Новизною** проведеного дослідження є аналіз методів та апаратних засобів підтримки технологій генерування випадкових послідовностей в системі криптографічного захисту рішень IoT; проведення класифікації атак на ГВЧ та особливостей захисту від них; визначення ефективних технологій та схемних рішень ГВЧ щодо використання в малопотужних пристроях IoT з обмеженими обчислювальними ресурсами; надання практичних рекомендацій щодо використання ГВЧ в системах криптографічного захисту мережі IoT. Аналіз технологій та схемних рішень дозволив сформувати наступні **висновки:** захист рішень IoT включає: безпеку вузлів мережі IoT та їх підключення до хмари за допомогою захищених протоколів, забезпечення конфіденційності, автентичності та цілісності даних в мережі IoT криптографічними методами, аналіз атак та моніторинг криптостійкості мережі IoT; первісною основою захисту рішень IoT є істинна випадковість послідовностей, які формуються ГВЧ і використовуються у алгоритмах криптографічного перетворення інформації для її захисту; особливістю пристроїв IoT є їх гетерогенність і географічний розподіл, обмеженість обчислювальних ресурсів та електроживлення, мініатюрність; найбільш ефективними (зменшують енергоспоживання та збільшують швидкість генерації) для застосування в пристроях IoT є RNG виключно на цифровій основі, в яких реалізується триступінчастий процес: початкова цифрова схема, нормалізатор та формувач потоку випадкових чисел; автономні булеві мережі (АБМ) дозволяють створити RNG з унікальними характеристиками: отримані числа є дійсно випадковими, висока швидкість – число можна отримати за один такт, мінімальне енергоспоживання, мініатюрність, висока (до 3 ГГц) пропускна здатність булевого хаосу; перспективним напрямом розвитку АБМ є використання оптичних логічних вентилів для побудови оптичних АБМ з пропускною здатністю до 14 ГГц; класифікація відомих класів атак на ГВЧ включає: прямі криптоаналітичні атаки, атаки, засновані на вхідних даних, атаки на основі розкриття внутрішнього стану ГВЧ, кореляційні атаки та спеціальні атаки; пакети статистичних тестів для оцінювання послідовностей RNG мають деякі обмеження або недоліки та не замінюють криптоаналіз; порівняння швидкодії криптоакселераторів з програмними засобами криптографічних перетворень показує їх значні переваги: для блокового алгоритму шифрування AES підвищується швидкодія в 10-20 разів у 8/16-бітових криптоакселераторах і в 150 разів у 32-бітових, хешування зростання швидкодії для алгоритмів SHA-1, SHA-256 у 32-бітових криптоакселераторів більш ніж в 100 разів, а для алгоритму HMAC – до 500 разів.

Ключові слова: інтернет речей; генератор випадкових чисел; криптостійкість; криптоаналіз; криптографічні ключі; шифрування; хешування; автономні бульові мережі; булевий хаос; статистичні тести; криптоакселератори.

КРИПТОСТОЙКИЕ МЕТОДЫ И ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ В УСТРОЙСТВАХ ИНТЕРНЕТ ВЕЩЕЙ (IOT)

Предмет исследования: криптостойкие методы и способы генерирования случайных последовательностей и аппаратная поддержка криптографических преобразований в устройствах IoT. **Целью** статьи является исследование криптостойких методов и средств генерирования и тестирования случайных последовательностей, пригодных для использования в IoT устройствах с ограниченными ресурсами; определение схемных реализаций аппаратных генераторов случайных последовательностей; формирование выводов по использованию генераторов случайных чисел (ГСЧ) в системах криптографической защиты сети IoT. В статье решаются следующие **задачи:** анализ методов и аппаратных средств формирования случайных последовательностей для защиты решений IoT с ограниченными ресурсами; определение безопасных и эффективных технологий реализации ГСЧ; классификация атак на ГСЧ; анализ недостатков практического использования пакетов статистических тестов для оценивания качества случайных последовательностей ГСЧ; оценивание быстродействия криптоакселераторов аппаратной поддержки криптографических преобразований; предоставление практических рекомендаций по ГСЧ для применения в устройствах IoT с ограниченными ресурсами. **Методы** исследования: метод структурно-функционального анализа ГСЧ и устройств IoT, криптографические методы защиты информации, методы генерирования случайных последовательностей, метод анализа устойчивости систем, методы построения автономных булевых сетей и анализа булевого хаоса, методы оценивания качества случайных последовательностей. **Результаты** работы: анализ технологий и схемных решений аппаратных ГСЧ по следующим характеристиками: качество случайности чисел и непредсказуемость последовательностей, быстродействие, энергопотребление, миниатюрность, возможность интегрального выполнения; предоставление практических рекомендаций по применению ГСЧ в системах криптографической защиты сети IoT. **Новизной** проведенного исследования является анализ методов и аппаратных средств поддержки технологий генерирования случайных последовательностей в системе криптографической защиты решений IoT; проведение классификации атак на ГСЧ и особенностей защиты от них; определение эффективных технологий и схемных решений ГСЧ по использованию в маломощных устройствах IoT с ограниченными вычислительными ресурсами; предоставление практических рекомендаций по использованию ГСЧ в системах криптографической защиты сети IoT. Анализ технологий и схемных решений позволил сформировать следующие **выводы:** защита решений IoT включает: безопасность узлов сети IoT и их подключение к облаку с помощью защищенных протоколов, обеспечение конфиденциальности, подлинности и целостности данных в сети IoT криптографическими методами, анализ атак и мониторинг криптостойкости сети; первоначальной основой защиты решений IoT является истинная случайность формируемых ГСЧ последовательностей и используемых в алгоритмах криптографического преобразования информации для ее защиты; особенностью устройств IoT является их гетерогенность и географическое распределение, ограниченность вычислительных ресурсов и электропитания, миниатюрность; наиболее эффективными (уменьшают энергопотребление и увеличивают скорость генерации) для применения в устройствах IoT являются ГСЧ исключительно на цифровой основе, в которых реализуется трехступенчатый процесс: начальная цифровая схема, нормализатор и формирователь потока случайных чисел; автономные булевые сети (АБС) позволяют создать ГСЧ с уникальными характеристиками: полученные числа действительно случайные, высокая скорость – число можно получить за один такт, минимальное энергопотребление, миниатюрность, высокая (до 3 ГГц) пропускная способность булевого хаоса; перспективным направлением развития АБМ есть использование оптических логических вентилей для построения оптических АБС с пропускной способностью до 14 ГГц; классификация известных классов атак на ГСЧ включает: прямые криптоаналитические атаки; атаки, основанные на входных данных; атаки на основе раскрытия внутреннего состояния ГСЧ; корреляционные атаки и специальные атаки; пакеты статистических тестов для оценивания последовательностей ГСЧ имеют некоторые ограничения или недостатки и не заменяют криптоанализ; сравнение быстродействия криптоакселераторов с программными средствами криптографических преобразований показывает их значительные преимущества: для блочного алгоритма шифрования AES повышается быстродействие в 10-20 раз в 8/16-битовых криптоакселераторах и в 150 раз – в 32-битовых, хеширования роста SHA-256 у 32-битных криптоакселераторов более чем в 100 раз, а для алгоритма HMAC – до 500 раз.

Ключевые слова: интернет вещей; генератор случайных чисел; криптостойкость; криптоанализ; криптографические ключи; шифрование; хеширование; автономные булевые сети; булевый хаос; статистические тесты; криптоакселераторы.

Бібліографічні описи / Bibliographic descriptions

Клімушин П. С., Соляник Т. М., Можаєв О. О., Гнусов Ю. В., Манжай О. В., Світличний В. А. Криптостійкі методи та генератори випадкових чисел у пристроях інтернет речей (IOT). *Сучасний стан наукових досліджень та технологій в промисловості*. 2022. № 2 (20). С. 22–34. DOI: https://doi.org/10.30837/ITSSI.2022.20.022

Klimushyn, P., Solianyk, T., Mozhaiev, O., Gnusov, Y., Manzhai, O., Svitlychny, V. (2022), "Crypto-resistant methods and random number generators in internet of things (IOT) devices", *Innovative Technologies and Scientific Solutions for Industries,* No. 2 (20), P. 22–34. DOI: https://doi.org/10.30837/ITSSI.2022.20.022